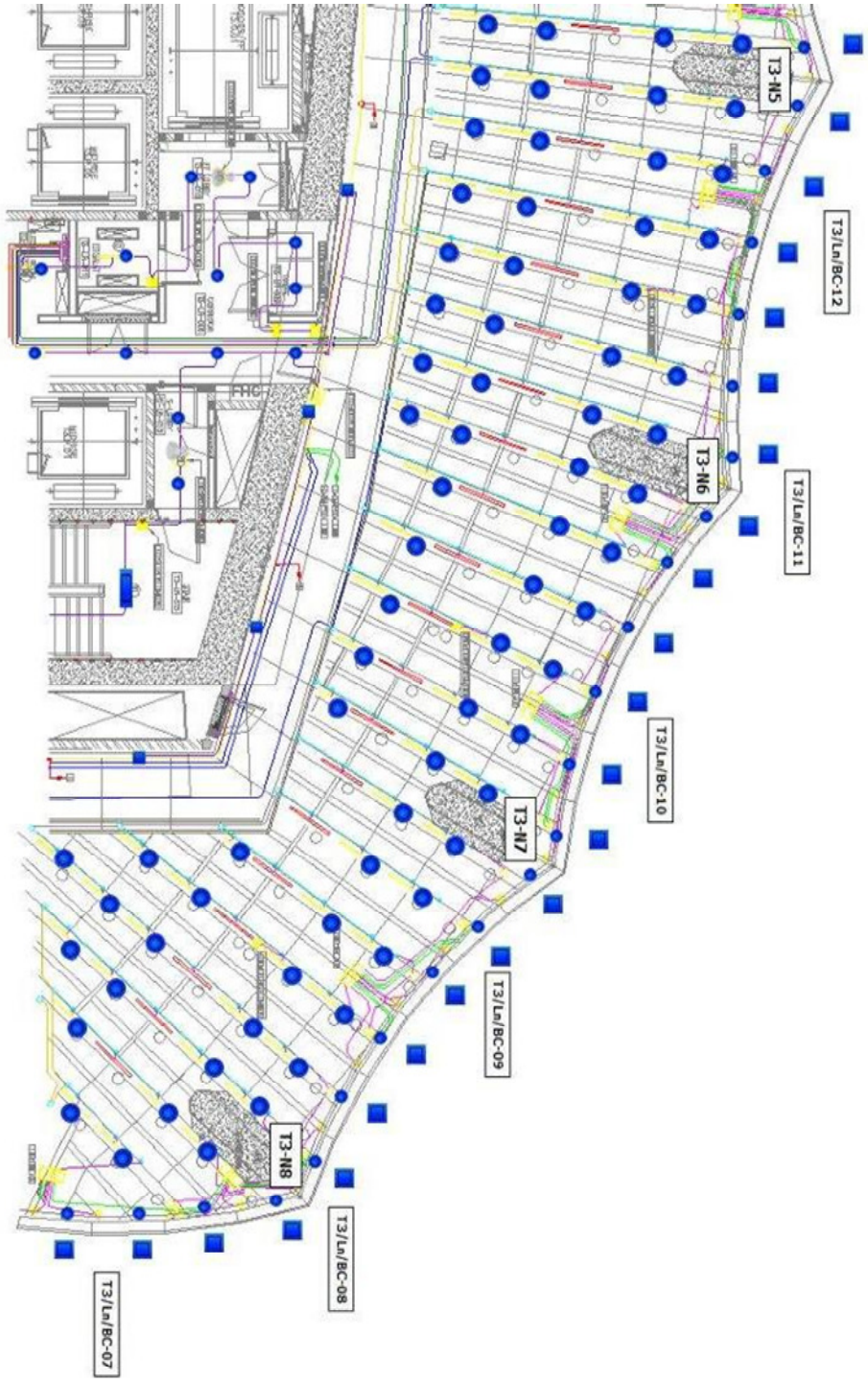
Graphical Widgets for Helvar IP Driver

Helvar

Installation and User Guide



Graphical Widgets for Helvar IP Driver

Table of Contents

1 Revision History 3
2 Document Outline 3
3 Scope 3
4 Safety Precautions 3
5 General Description 3
6 System Requirements 4
7 Disclaimer 4
8 Installation 4
9 Widget Overview 5
10 Adding a Widget onto a PX file 5
10.1 Adding a widget using Automatic Binding 6
10.2 Adding a widget using Manual Binding 7
11 Graphical Widget Component Guides 8
11.1 PIR Sensor Widget 8
11.2 ScenePlate Widget 9
11.3 Luminaire Widget 11
11.4 Emergency Luminaire Widget 15
12 Component Creator Widget Guide 19

1 Revision History

Issue	Date	Change Descriptions	Author
Issue 01	18/03/13	First Issue	Helvar Limited

2 Document Outline

This document is the software manual for version 1.0 of the Graphical Widgets developed for the Helvar IP Driver, Tridium Jace hardware and Ax Supervisor using the Niagara AX platform.

3 Scope

This document is for use by system integrators (SI's) that belong to the Tridium partner network and have completed training on the use of the Niagara AX Framework and Tridium hardware.

4 Safety Precautions

None.

5 General Description

This document describes the Graphical Widgets for Niagara AX. This is used in addition to the Helvar IP Driver and Helvar Digidim routers to provide an enhanced graphical user interface for Lighting control based on the Niagara Framework.

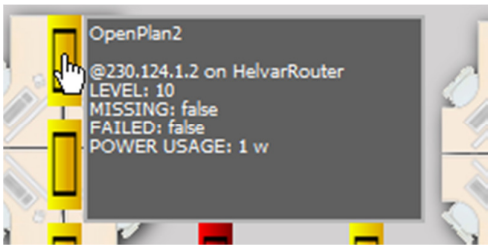


Figure 1: Luminaire Widget with hover over pop-up

Using the graphical widgets will greatly increase the productivity of an SI when building PX pages. There are four graphical widgets representing each of the common component types found in a lighting system. Each widget provides:

- Pre-defined graphics that represent the state of the lighting component e.g On, Off, Fault
- For luminaire widgets, a 'mouse hover over' feature that displays the current operating parameters of the lighting component e.g. Address, Level, Status
- The ability, where appropriate, to directly control the lighting component e.g. Toggle On/Off
- The ability to set the properties of each graphical widget to suit the particular application

In addition there is a Component Creator widget that automates the process of adding runtime strategy for all luminaire objects. This can be exposed in the luminaire graphical widgets to display the total lamp burning hours. Its use and operation are covered in Section 12.

6 System Requirements

This widgets have been tested with the Helvar IP driver, Niagara AX platform version 3.7.X and Tridium JACE model JCX 660. The widgets requires the use of Helvar Lighting Router firmware version 4.2.16 or later.

Widgets can operate directly on a Jace/Supervisor running the Station/PX pages or by virtuals over the Niagara network.

The widgets can also be hosted on a PC based station, without the use of a JACE, running under the Niagara AX platform.

7 Disclaimer

Helvar Ltd. ("Helvar") has developed this graphical widget software for use with the Helvar IP driver operating with the Tridium, Inc. (“Tridium”) ‘Niagara AX Framework®’. By installing this widget software you accept the terms and conditions of this disclaimer. Furthermore if you are using this software on behalf of, or for the benefit of, an end user, then you also represent that you are authorized by the end user to accept the terms and conditions of this disclaimer for the end user as the agent of the end user. If you do not agree to these terms, or if you are not authorized by the end user to accept the terms of this disclaimer for the end user, then this software should be removed immediately.

Helvar provides this graphical widget software "as is," and makes no express or implied warranty of any kind. Helvar specifically disclaims all indirect or implied warranties to the full extent allowed by applicable law, including without limitation all implied warranties of, non-infringement, merchantability, title or fitness for any particular purpose. No oral or written information or advice given by Helvar, its agents or employees shall create a warranty.

Neither Helvar nor any of its officers, agents, employees or representatives shall be liable for indirect, special, incidental, consequential, or punitive damages or penalties (including damages for lost profits, lost business, personal injury, lost data, business interruption, and the like), however it arises, whether for breach or in tort, even if Helvar has been advised of the possibility of such damages.

Helvar standard terms and conditions apply.

8 Installation

The Helvar Graphical Widgets are installed in the modules directory of Niagara AX Workbench and can subsequently be configured into a Tridium Station (JACE unit or AX Supervisor). The files are named ‘helvarWidget.jar’ and ‘helvarComponentCreator.jar’.

The Helvar Grapical Widgets are not licensed, but will work only with a licensed version of the HLC-IPDRV Helvar IP driver.

Once installed Workbench and stations should be re-started.

9 Widget Overview

The Helvar Widget module contains 4 basic graphical components for use in PX graphics:

- Scene Plate Widget - a graphical PX widget to represent a Scene Plate.
- Luminaire Widget – a graphical PX widget to represent a Luminaire.
- Emergency Luminaire Widget – a graphical PX widget to represent a DALI Emergency Luminaire.
- PIR Widget – a graphical PX widget representing a PIR sensor, this is used to display the occupied/unoccupied status of an area.

Each of these components have a number of variations available in the helvarWidget palette.

10 Adding a Widget onto a PX file

Once the HelvarWidget module has been installed the widgets are accessible by opening the helvarWidget Palette.

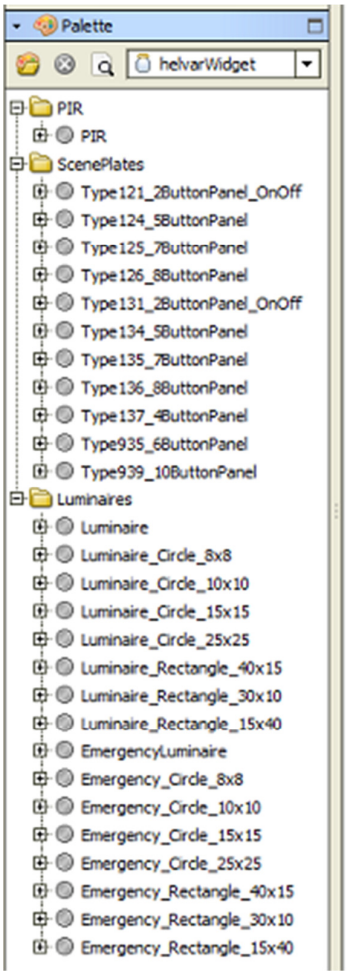


Figure 2: Helvar Widget Palette

The palette is organized into three folders: PIR, ScenePlates and Luminaires. Within each folder are one or more variations of the particular widgets catering for different ScenePlate configurations, and various sizes and shapes of luminaires and emergency luminaires. The Luminaires folder contains both luminaire widgets and emergency luminaire widgets.

Widgets can simply be dragged onto PX files directly from the palette and their bindings be set manually, alternatively you can drag the appropriate Helvar points from the Workplace Nav tree onto a PX file and have the bindings configured automatically. The automatic binding method considerably reduces the engineering time of creating graphics.

To create widgets and take advantage of automatic binding each type of widget component has an associated point type within the Helvar Network:

- For PIR Sensors – the occupancy point under the points folder of the PIR input device, in the particular router.
- For ScenePlates – the Group component under the Groups folder in the HelvarNetwork.
- For Luminaires – the OutputLevel point under the points folder of the Lumiaire output device in the particular router.
- For Emergency Luminaires – the EmergencyFunctionTestTime point under the points folder of the EmergencyLuminaire output device in the particular router.

10.1 Adding a widget using Automatic Binding

To take advantage of automatic binding of the Helvar Widgets drag the associate point onto a PX file. The Make Widget dialog will appear:

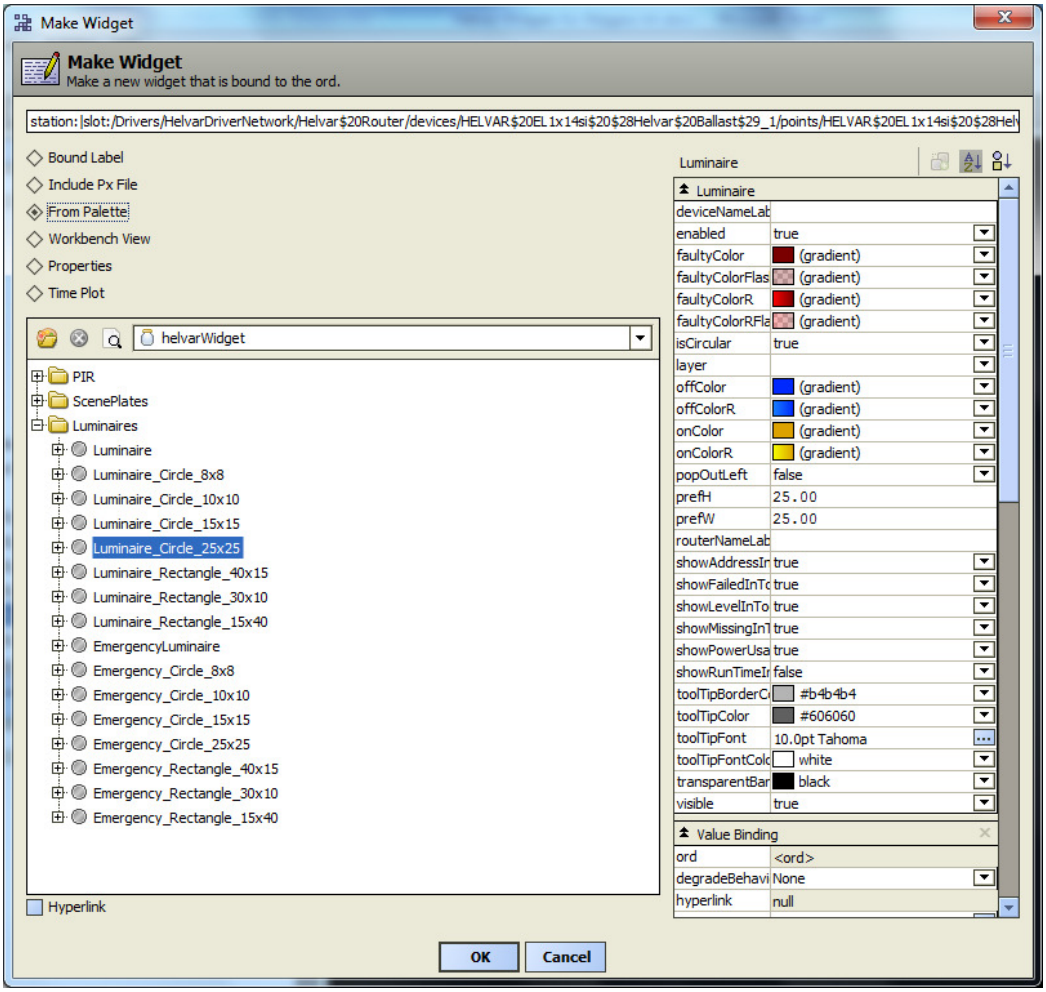


Figure 3: Make Widget Dialog

In the options in the top left of the dialog ensure that From Palette is checked. Open the helvarWidget palette and select the required variant of widget component from the palette. In the component view on the right side of the dialog the top-most value binding should have its ord property set to <ord>. Click OK to create and bind the widget instance.

10.2 Adding a widget using Manual Binding

To create a widget and manually set the value bindings, simply drag the desired widget from the palette onto the PX graphic. Double-click on the widget to configure its properties and value bindings. Select the top-most value binding and configure its ord property:

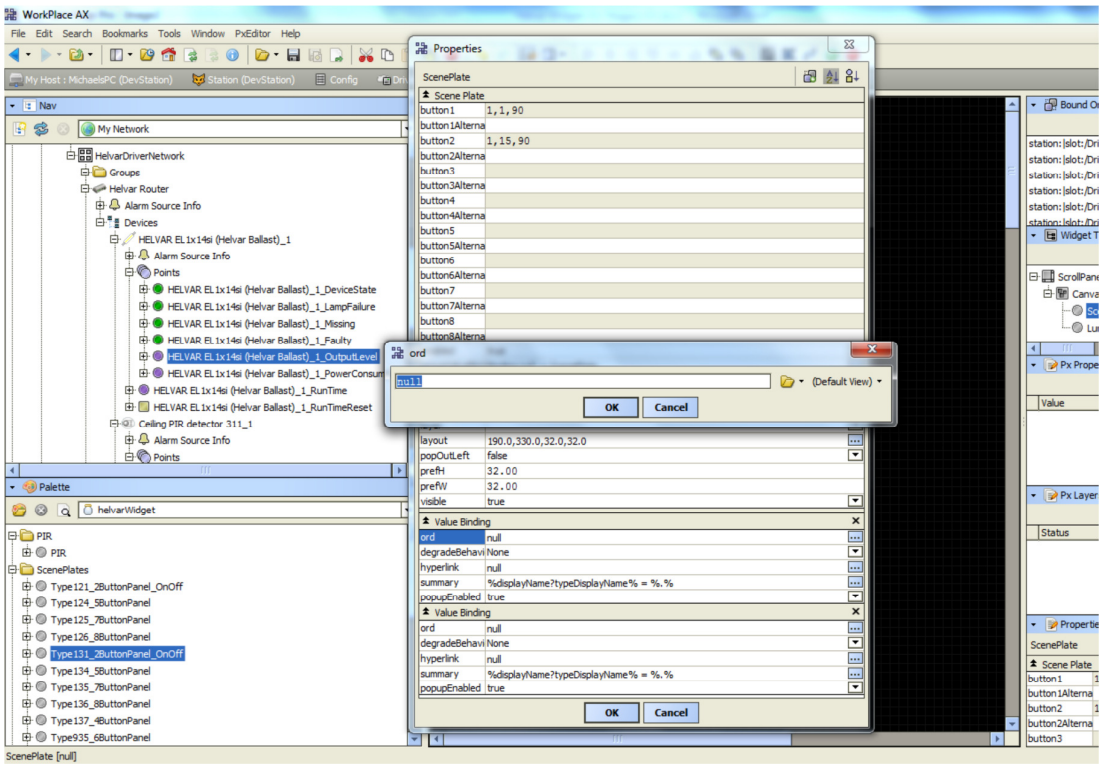


Figure 4: Editing the base ord

Providing the ord is set the the correct path for the type of widget, all the other value bindings will be set automatically once the Properties dialog has been closed down using the OK button.

N.B. The ord path must be set to the correct point for each type of widget as detailed in Section 10, otherwise the automatic binding of all but the top-most value binding will fail.

11 Graphical Widget Component Guides

11.1 PIR Sensor Widget

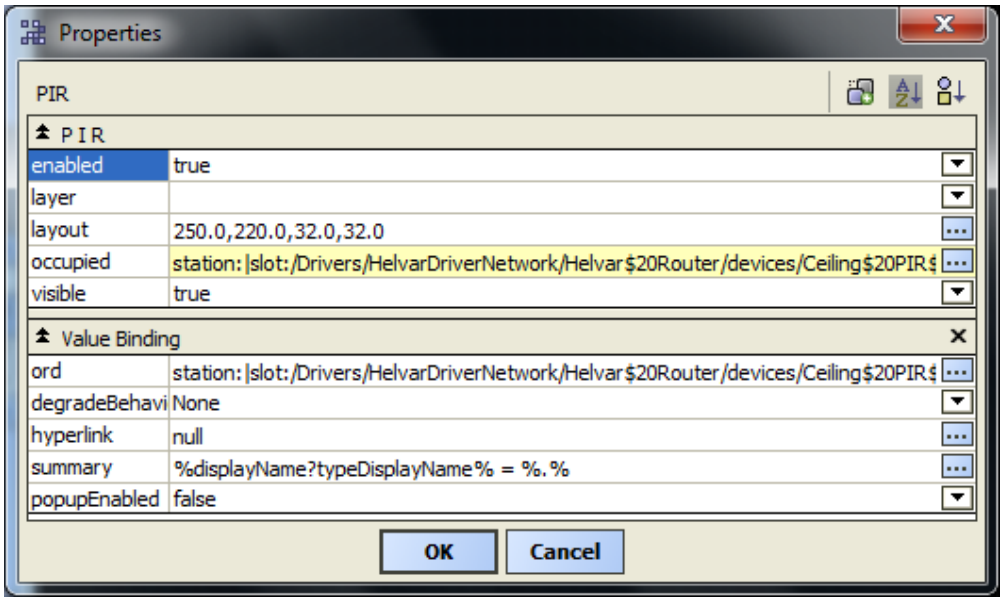
The PIR (Passive InfraRed) sensor widget is bound to an occupancy sensor in the lighting system and is used to represent the occupancy status of an area.

Occupied

Unoccupied

It should be noted that an 'Occupied' state does not guarantee that a person is currently present. This is due to system latency when changing from an 'Occupied' to 'Unoccupied' state and is due to sensor timeouts.

It has the following properties:

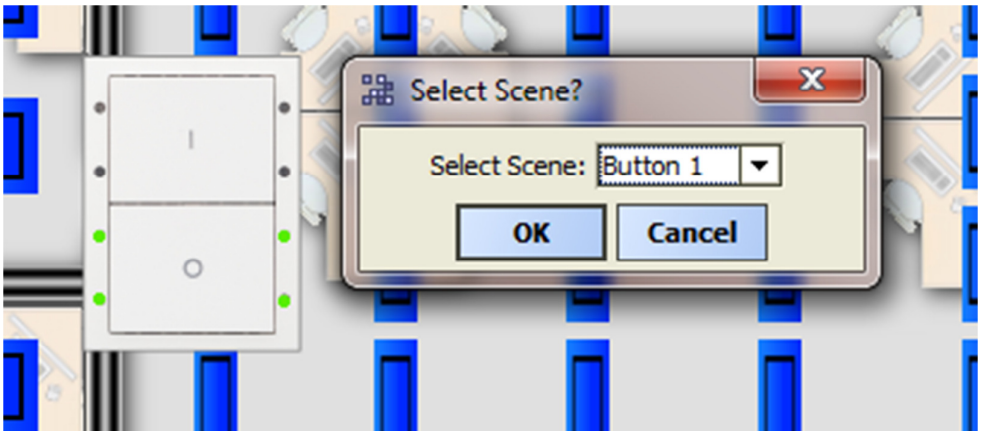


Occupied	This Boolean property is the occupancy property of the corresponding PIR sensor. It should be bound to the Value binding by default. This property should not be altered.
Value Binding	this is used to relate the widget to a particular occupancy sensor. If the value binding has not been automatically configured, the ord property should be set to the path of the occupancy point under the points folder of the PIR input device, in the particular router.

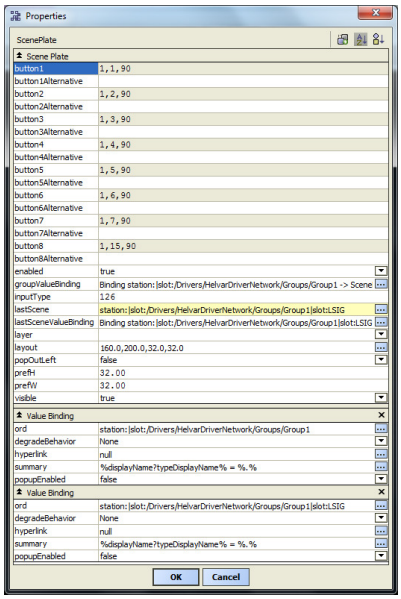
11.2 ScenePlate Widget

The Scene plate widget is used to represent a scene plate and allows the user to view and select the current scene for a particular group. When the px page is in its normal view mode the Scene plate widget appears as an icon that represents a particular panel type.

Hovering the mouse over the icon expands the graphic of the scene plate, showing the LED tellback for the currently selected scene. Left clicking on the expanded graphic pops up a dialog that allows a button to be selected and actioned. The command issued to the lighting system is defined by the properties set for that button.



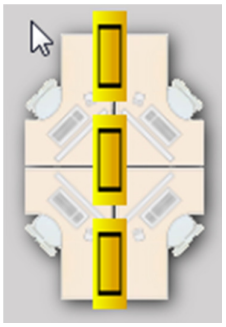
The properties available depend upon the selected scene plate widget. Those described below are for an 8-button panel type:



Button 1 - 8	These read-only properties shows the block, scene and fade time for each button command separated by commas. These properties are set by the system according to the value of the inputType property.
Button1Alternative – Button8Alternative	These properties allow the engineer to override the default scene selection commands for each button.
InputType	the Helvar Input type number. Altering this manually will result in modified button commands. The inputType is also used to determine which scene plate image is displayed for this instance of the widget. Valid values are: 121, 124, 125, 126, 131, 132, 134, 135, 136, 137, 935, 939.
Value Binding 1	This is used to bind the widget to a particular group. If the value binding has not been automatically configured, the ord property should be set to the path of the Group component under the Groups folder in the HelvarNetwork. Once this property has been set correctly the Value Binding 2 will be configured automatically by clicking on OK.
Value Binding 2	this is used to bind the widget to the Last Scene In Group property of a Group (LSIG). This property will be set automatically when Value Binding 1 is set correctly.

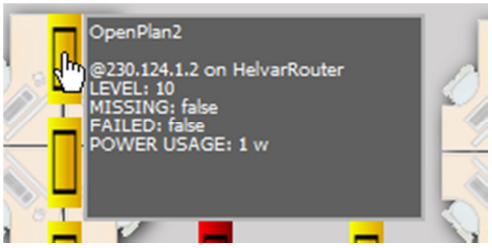
11.3 Luminaire Widget

The Luminaire widget is used to represent a non-emergency luminaire and allow the user to set the luminaire to a direct level of 0, or 100. When the px page is in its normal view mode the Luminaire widget appears as an icon. The size, shape and colour of the icon can be adjusted in the widget's properties. However the widget palette contains a number of icon choices with a predefined shape and size.

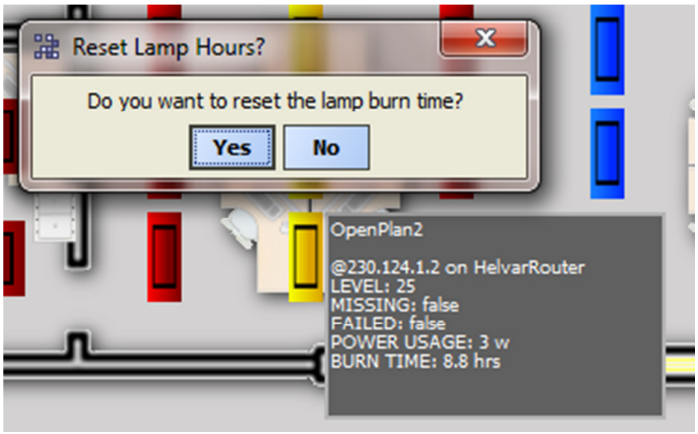


Left-clicking on the icon toggles the luminaire state between On/Off.

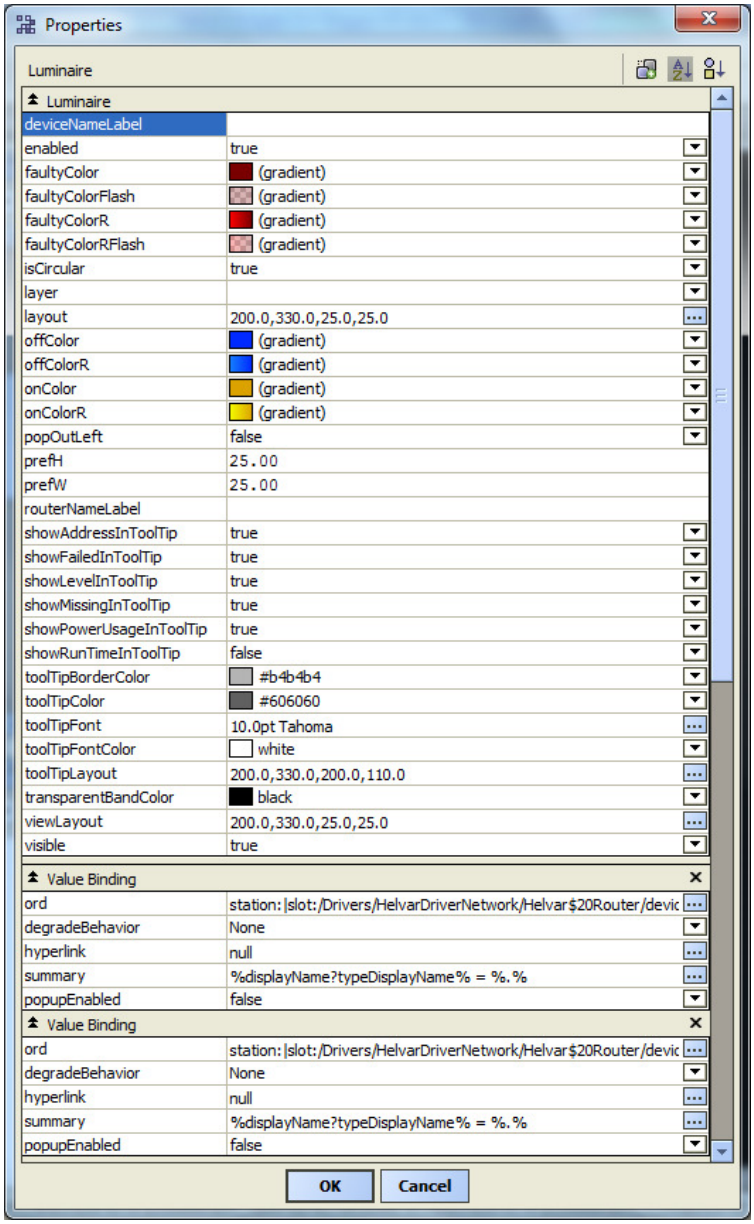
Hovering the mouse over the icon pops up an information text box showing the device parameters that have been set as visible in the properties.



Right-clicking on the icon pops up a dialog that allows the device's RunTime value to be reset. This option appears only if the 'shownRunTimeInTooltip' property has been set to TRUE and the Component creator widget has been used to create the required strategy and points for the device (See Section 12).



A Luminaire widget has the following properties:



deviceNameLabel	This property is used to set the device name shown on the luminaires tooltip. If this is left blank then the system will automatically set the device name.
faultyColor	The colour of the widget when the luminaire is faulty. This property is only used for circular instances of the widget.

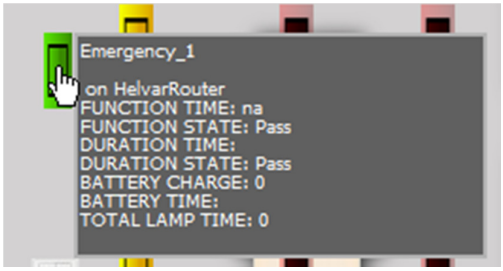
faultyColorFlash	The alternate flash colour of the widget when the luminaire is faulty. This property is only used for circular instances of the widget.
faultyColorR	The colour of the widget when the luminaire is faulty. This property is only used for rectangular instances of the widget.
faultyColorRFlash	The alternate flash colour of the widget when the luminaire is faulty. This property is only used for rectangular instances of the widget.
IsCircular	Determines if this luminaire widget is circular or rectangular.
offColor	The colour of the widget when the luminaire is off. This property is only used for circular instances of the widget.
offColorR	The colour of the widget when the luminaire is off. This property is only used for rectangular instances of the widget.
onColor	The colour of the widget when the luminaire is on. This property is only used for circular instances of the widget.
onColorR	The colour of the widget when the luminaire is on. This property is only used for rectangular instances of the widget.
popOutLeft	Used to determine if the luminaire widget's tooltip pops out to the left (true) or to the right (false).
prefH	The height of the widget icon.
prefW	The width of the widget icon.
routerNameLabel	This property is used to set the router name shown on the luminaires tooltip. If this is left blank then the system will automatically set the router name.
showAddressInTooltip; showFailedInTooltip; showLevelInToolTip; showMissingInToolTip; showPowerUsageInTooltip; shownRunTimeInTooltip	, showFailedInTooltip, showLevelInToolTip, showMissingInToolTip, showPowerUsageInTooltip, shownRunTimeInTooltip – these properties are used to determine what information about the luminaire is shown in its tooltip.
toolTipBorderColor	The colour of the luminaires tooltip border.
toolTipColor	The background colour of the luminaires tooltip.
toolTipFont	The font used on the tooltip.
toolTipFontColor	The font colour used on the tooltip.
toolTipLayout	The size of the luminaires tooltip.
transparentBandColor	The colour used on the band drawn on the luminaire icon. This it is intended that this be set to the background colour.

Value Binding 1	This is used to bind the widget to a particular luminaire's output level. If the value binding has not been automatically configured, the ord property should be set to the OutputLevel point under the points folder of the Lumiaire output device in the particular router. Once this property has been set correctly the Value Bindings 2 - 8 will be configured automatically by clicking on OK.
Value Binding 2	This is used to bind the widget to the luminaires Missing status point. This property will be set automatically when Value Binding 1 is set correctly.
Value Binding 3	This is used to bind the widget to the luminaires LampFailure status point. This property will be set automatically when Value Binding 1 is set correctly.
Value Binding 4	This is used to bind the widget to the luminaires PowerConsumption status point. This property will be set automatically when Value Binding 1 is set correctly.
Value Binding 5	This is used to bind the widget to the luminaires Address property. This property will be set automatically when Value Binding 1 is set correctly.
Value Binding 6	This is used to bind the widget to the luminaires RunTime status point. This status point only exists if the HelvarComponentCreator module has been used to add this status point. This property will be set automatically when Value Binding 1 is set correctly.
Value Binding 7	This is used to bind the widget to the luminaires RunTimeReset command point. This command point only exists if the HelvarComponentCreator module has been used to add this point. This property will be set automatically when Value Binding 1 is set correctly.
Value Binding 8	This is used to bind the widget to the luminaire device. This property will be set automatically when Value Binding 1 is set correctly.

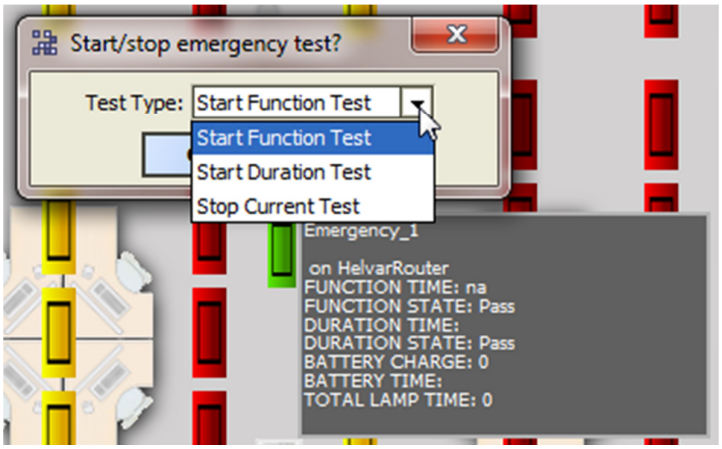
11.4 Emergency Luminaire Widget

The Emergency Luminaire widget is used to represent an emergency luminaire and allow the user to start and stop the emergency luminaire's function and duration tests.The size, shape and colour of the icon can be adjusted in the widget's properties. However the widget palette contains a number of icon choices with a predefined shape and size.

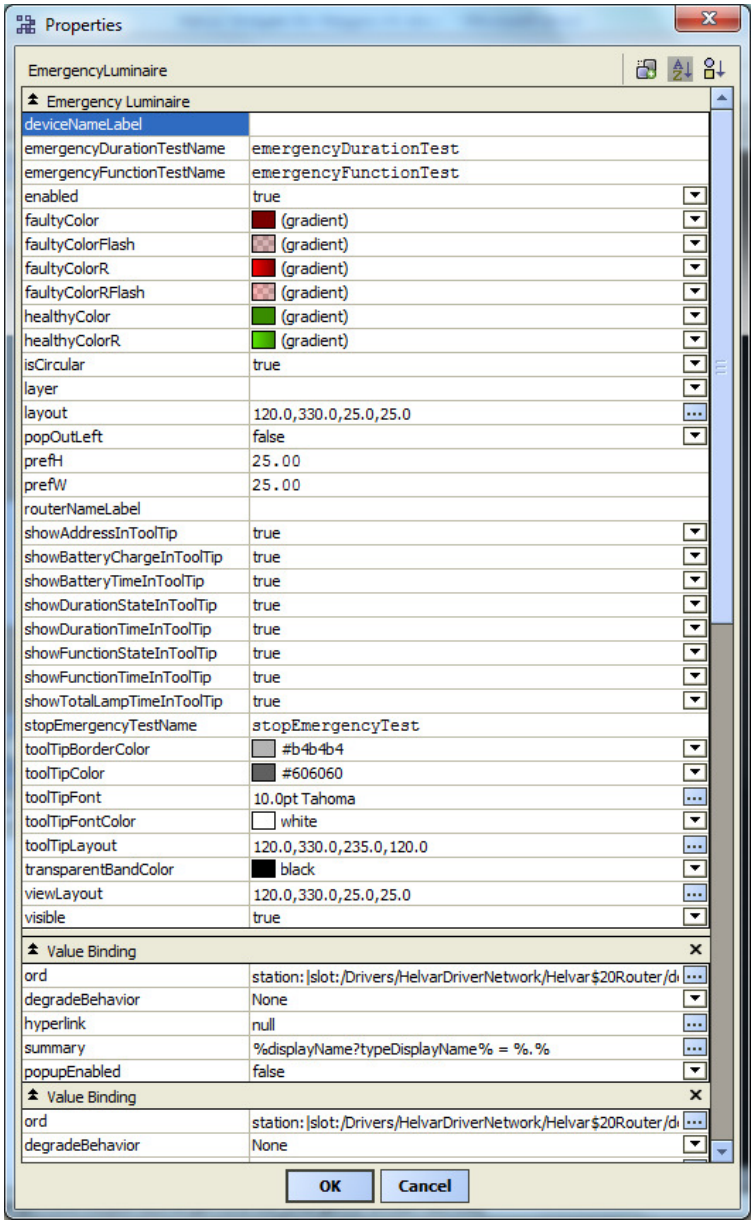
Hovering the mouse over the icon pops up an information text box showing the device parameters that have been set as visible in the properties.



Left/Right-clicking on the icon pops up a dialog that allows the user to select an emergency test type or stop the current test in progress.



An Emergency luminaire widget has the following properties:



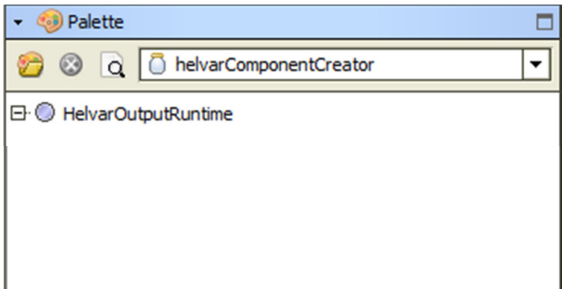
deviceNameLabel	This property is used to set the device name shown on the luminaires tooltip. If this is left blank then the system will automatically set the device name.
faultyColor	The colour of the widget when the luminaire is faulty. This property is only used for circular instances of the widget.

faultyColorFlash	The alternate flash colour of the widget when the luminaire is faulty. This property is only used for circular instances of the widget.
faultyColorR	The colour of the widget when the luminaire is faulty. This property is only used for rectangular instances of the widget.
faultyColorRFlash	The alternate flash colour of the widget when the luminaire is faulty. This property is only used for rectangular instances of the widget.
IsCircular	– determines if this luminaire widget is circular or rectangular.
offColor	The colour of the widget when the luminaire is off. This property is only used for circular instances of the widget.
offColorR	The colour of the widget when the luminaire is off. This property is only used for rectangular instances of the widget.
onColor	The colour of the widget when the luminaire is on. This property is only used for circular instances of the widget.
onColorR	The colour of the widget when the luminaire is on. This property is only used for rectangular instances of the widget.
popOutLeft	– used to determine if the luminaire widget's tooltip pops out to the left (true) or to the right (false).
prefH	The height of the widget icon.
prefW	The width of the widget icon.
routerNameLabel	This property is used to set the router name shown on the luminaires tooltip. If this is left blank then the system will automatically set the router name.
showAddressInTooltip	, showFailedInTooltip, showLevelInToolTip, showMissingInToolTip, showPowerUsageInTooltip, shownRunTimeInTooltip – these properties are used to determine what information about the luminaire is shown in its tooltip.
toolTipBorderColor	The colour of the luminaires tooltip border.
toolTipColor	The background colour of the luminaires tooltip.
toolTipFont	The font used on the tooltip.
toolTipFontColor	The font colour used on the tooltip.
toolTipLayout	The size of the luminaires tooltip.
transparentBandColor	The colour used on the band drawn on the luminaire icon. This it is intended that this be set to the background colour.

Value Binding 1	This is used to bind the widget to a particular luminaire's output level. If the value binding has not been automatically configured, the ord property should be set to the OutputLevel point under the points folder of the Lumiaire output device in the particular router. Once this property has been set correctly the Value Bindings 2 - 8 will be configured automatically by clicking on OK.
Value Binding 2	This is used to bind the widget to the luminaires Missing status point. This property will be set automatically when Value Binding 1 is set correctly.
Value Binding 3	This is used to bind the widget to the luminaires LampFailure status point. This property will be set automatically when Value Binding 1 is set correctly.
Value Binding 4	This is used to bind the widget to the luminaires PowerConsumption status point. This property will be set automatically when Value Binding 1 is set correctly.
Value Binding 5	This is used to bind the widget to the luminaires Address property. This property will be set automatically when Value Binding 1 is set correctly.
Value Binding 6	This is used to bind the widget to the luminaires RunTime status point. This status point only exists if the HelvarComponentCreator module has been used to add this status point. This property will be set automatically when Value Binding 1 is set correctly.
Value Binding 7	This is used to bind the widget to the luminaires RunTimeReset command point. This command point only exists if the HelvarComponentCreator module has been used to add this point. This property will be set automatically when Value Binding 1 is set correctly.
Value Binding 8	This is used to bind the widget to the luminaire device. This property will be set automatically when Value Binding 1 is set correctly.

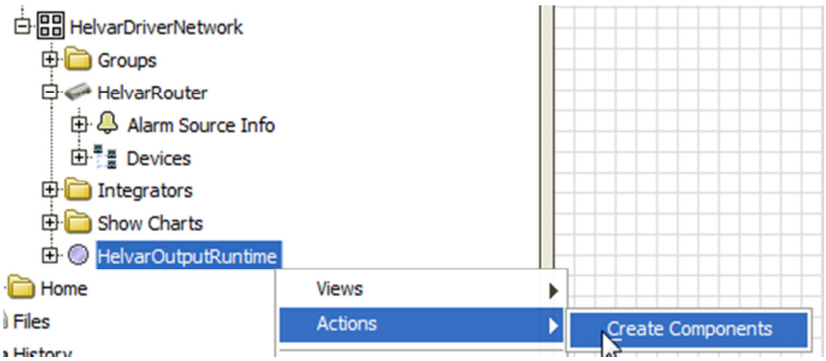
12 Component Creator Widget Guide

From the Palette select the ‘helvarComponentCreator’ widget from the drop down list or use the open dialog to search for the widget. Once open the Palette will show one item the ‘HelvarOutputRuntime’.

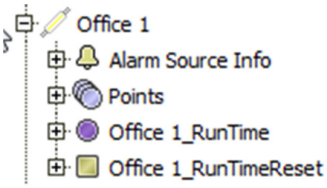


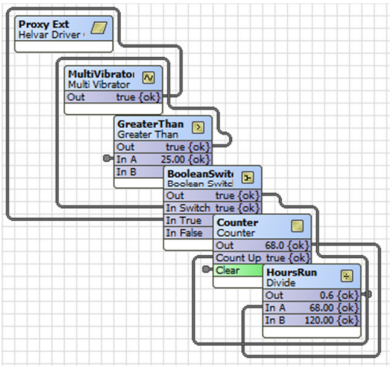
Select this item and drag it on to the ‘HelvarDriverNetwork’ in the Nav tree. OK the name dialog and the ‘HelvarOutputRuntime’ will appear in the tree.

Right-click the ‘HelvarOutputRuntime’ in the Nav tree and select Actions►Create Components. This may take several minutes to complete, depending on the number of devices with OutputLevel points.



Once the process is complete, strategy will have been added to the wire sheet for each OutputLevel point. In addition two new items RunTime and RunTimeReset will appear in the tree at device level.





The new strategy and points can now be exposed in the Luminaire widget to display the total run time for the luminaire. By default the Luminaire and Emergency Luminaire widgets do not display the run time total. To make this item visible set the ‘showRunTimeInToolTip’ property to TRUE (see Section 11.3)